Computational Methods for Investigating Higher Teichmüller Spaces

November 30, 2020

Presented by Arjun Malik.

In partial fulfillment of the requirements for graduation with the Dean's Scholars Honors Degree in Mathematics.

Supervising Professor: Florian Stecker	Date
Second Reader 1: Lorenzo Sadun	Date
Second Reader 2: Oscar Gonzales	Date

1 Abstract

This paper deals with the discrete subgroup problem, solvable through the study of higher Teichmüller theory. The discrete subgroup problem is as follows: given a Lie group (a group with continuity) such as $\mathbb{R}$, how do you find discrete subgroups, such as $\mathbb{Z}$, particularly those of a type similar to (isomorphic to) a certain group? This paper details how, using theorems pertaining to Higher Teichmüller theory graphing the eigenvalue gaps of a matrix group can be used to figure out if it is a discrete subgroup of a Lie Group. In it, I find two subsets of results: one is working towards an attempt to validate recently found results about a Lie group, and the other is a result about the method itself (specifically, projective convex bending) and how it can be used empirically.

2 Background

2.1 Motivation

I feel it is important to start with one major motivation for studying higher Teichmüller theory, explained in a simpler way than the rest of the background section: the problem of finding discrete subgroups of a Lie group. A Lie group is a group that can have a notion of continuity and in which taking products and inverses of elements are smooth, or differentiable, as operations with this notion. A discrete subgroup of a Lie group is one where every point is isolated under the topology on the group. For instance, $\mathbb{R}$ can be considered as a Lie group, with addition being the group operation. A discrete subspace of this is $\mathbb{Z}$, using the standard topology. Now, how would one find all the discrete subgroups? Well, one could look at the maps to the space from a group that is already discrete. Finding all discrete subgroups isomorphic to a certain discrete group (like $\mathbb{Z}$) takes some thought. For instance, consider the group S^1, the circle group. What are the discrete subgroups of this that are isomorphic to $\mathbb{Z}$, using the natural map from $\mathbb{R}$ to the circle group (where we consider the circle group as just $\mathbb{R}$, with the relation that $2\pi = 0$)? In this case it is pretty simple, but it does require some knowledge of the circle group. Each group isomorphic to $\mathbb{Z}$ has one generator and no relations. If this generator is a multiple of 2π by a rational number a, it will eventually reach the original number plus an integer multiple of 2π, which is just the original number. This subgroup is discrete, and there exists a homomorphism from $\mathbb{Z}$ to it, but is not isomorphic as that map is not injective or invertible. Else, it will never reach the original number, and in fact maps to a dense subset of S^1. Therefore, it is not a discrete subgroup. Therefore, there are no discrete subgroups of the circle group isomorphic to $\mathbb{Z}$.

This example was easy to solve, but it shows that the problem of finding discrete subgroups of a group, even ones that can be found by isomorphisms from a certain group, is not a trivial problem in general. In order to understand the version of the problem in this paper, I will now introduce the fundamental group of a surface, also known as a surface group, as well as a few commonly studied Lie groups. Every closed orientable surface (surfaces that are the outsides of solids) has an associated group that distinguishes it from other different such surfaces, up to homeomorphism. This group is generated by the sides of the shape that can be folded to create a surface of that genus, with a relation such that going the whole way around the shape, or the surface, gives the identity, as it should. For instance, the surface group for the torus is generated by a and b, representing the sides of the square, and has the relation $aba^{-1}b^{-1} = e$. For a general surface group, this is a_1 through a_g and b_1 through b_g, with relation $a_1b_1a_1^{-1}b_1^{-1}a_2b_2a_1^{-1}b_2^{-1}...a_gb_ga_g^{-1}b_g^{-1} = e$. I chose to call the index g because it is the genus of a surface, which is fundamentally, for closed orientable surfaces, the number of holes it has. [5]

Some basic examples of Lie groups are $GL(n, \mathbb{R})$, the group of real $n \times n$ matrices, $SL(n, \mathbb{R})$, the subgroup of $GL(n, \mathbb{R})$ where every matrix has determinant 1, $O(n)$, which is the set of $n \times n$ matrices whose transpose is their inverse, and $SO(n)$, the subset of $O(n)$ where every matrix has determinant 1. Other relevant lie groups involve $PSL(n, \mathbb{R})$, which is $SL(n, \mathbb{R})$ except the negative of a matrix is defined as being equal to it, and $SO(p, q)$. We can view $SO(n)$ as a set of matrices A, with determinant 1, where $AIA^T = I$, the identity matrix in the middle being superfluous. SO(p, q) extends this concept to a set of matrices A, with determinant 1, where $AJA^T = J$, where J has p 1s on the diagonal and q -1s. These matrices have dimension $n \times n$.

In addition, for a countable group, such as the discrete groups and subgroups of interest, you can represent the elements as words. Each element of the group is a product of some generators, by the definition of a group. Writing a group element in terms a product of generators is known as writing it in terms of a word. Due to the group relations, there are several equivalent ways of writing a word. The shortest way is known as a reduced word, and the length of this word is known as the word length of the element.

However, for a certain case of finding discrete subgroups of $PSL(2, \mathbb{R})$ with homomorphisms from a surface group, there does exist a neat solution. Specifically, if we define a topology on the variety of representation, that variety can be shown to split up into $4g - 3$ connected components, where g is the genus of the surface. Two of these connected components form the Teichmüller space. The representations in this component are all discrete and injective, and therefore the groups in the image of these representations (called the Fuchsian groups) are discrete subgroups of the space. (Throughout the paper, I'll use the phrase "representations in a Teichmüller space," despite the actual elements being representations up to conjugation. This will refer to one single representative representation in the set of representations that comprises the actual element.)

2.2 Higher Teichmüller Spaces

It is best to restate this as a formal definition before going into Higher Teichmüller Spaces. A Teichmüller space of a hyperbolic surface, in the definition most relevant to the concept of a higher Teichmüller space, is a subset of the variety of representations from the fundamental group of the surface, which we'll call $\pi(S)$ into PSL(2, R), up to conjugation by PSL(2, R). Specifically it is a union of two of the connected components of this variety, under a topology induced from PSL(2, R). These components are specifically the one that contain discrete and injective components. A higher Teichmüller space is a generalization that simply replaces PSL(2, R) with a different group G. There are two known types of higher Teichmüller spaces, meaning two types of groups G for which there is a connected component of the representation variety with discrete and faithful representations within it, and other nice properties.

The first, called the Hitchin component, consists of specifically split real semisimple Lie groups. This concept is beyond the scope of this paper, but relevant examples among the Lie groups we've listed are SL(n, $\mathbb{R}$) and SO(n, n+1)[9]. These are defined by first taking any discrete and faithful representation from $\pi(S)$ to SL(2, $\mathbb{R}$), and then composing them with a unique embedding of SL(2, $\mathbb{R}$) into the Lie group that exists for every split real semisimple Lie group, called the three-dimensional principal subgroup. These representations are explained in greater detail below. The Hitchin component is the connected component of the representation variety containing this representation. The Hitchin component containing homomorphisms from a surface group of genus 2 into $SL(3, \mathbb{R})$ was used as a test higher Teichmüller space for the program described in this paper. It has the property that the image of every element of the surface group under a representation in the Hitchin component is diagonalizable, with distinct eigenvalues of the same sign, and that every representation within is discrete and faithful [4].

The second, called the space of maximal representations, are defined when simple Lie groups are of Hermitian type. Again, this concept can be demonstrated by examples of groups of such type, like $PSL(2, \mathbb{C})$ and $SO(2, n)$ It is essentially defined by assigning a characteristic number, called the Toledo number, to these representation, and finding the set of representations where this number is maximized. This ends up being a union of connected components. They have similar properties to the Hitchin representations above. Maximal representations are unimportant for this paper: for more on them you can check out Anna Wienhard's paper on Higher Teichmüller Theory in general: [9].

The paper "Positivity and higher Teichmüller theory" is most notable for proposing that the groups SO(p, q), where p is not equal to q, and the exceptional family modeled on F_4 also admit higher Teichmüller spaces, meaning there is a connected component of representations from the fundamental group of a surface to this that are discrete and faithful. Fundamentally this paper claims that the concept of positivity can be extended to representations in such a way that those representations in a higher Teichmüller space are "positive" in some way, and that there are other representations positive in the same way, into $SO(p,q)$ and that exceptional family.

2.3 p-dominated representations and computational tests

A paper by Bochi, Potrie, and Sambarino shows that a certain property[1] of representations contained within higher Teichmüller spaces are p-dominated representations for all p. These representations can be defined as follows. Let any homomorphism ρ map any element of the surface group a to its image in the matrix Lie group A, with eigenvalues $\lambda_1, ..., \lambda_n$. In this case, for a given p, $\frac{\lambda_p}{\lambda_{p-1}} \geq Ce^{l|a|}$, where $|a|$ is the word length of a, for some constants C and $l > 0$. All representations in higher Teichmüller spaces with are p-dominated for $p \in \{1, .., n-1\}$[3] You can easily derive from this that if you were to graph, for a 3×3 matrix, $\frac{\lambda_3}{\lambda_2}$ against $\frac{\lambda_2}{\lambda_1}$, each individual element would have to have a vertical coordinate close to its horizontal coordinate, as every eigenvalue gap is bounded like this. If you were to do this for any set of matrices in the image of ρ, then the entire plot would be bounded below and above by exponential functions, or on a log graph, be bounded above and below by straight lines, much like a 2-D cone. Therefore, if you graph a set of matrices in the image of a representation and the resulting scatter plot is not bounded in such a way, barring other errors, that representation cannot be p-dominated, which means a space that contains it is not a higher Teichmüller space. This however, requires us to choose multiple representations ρ that we can test, and find the matrices in the image of them. The rest of the paper will be about the choice of ρ and the process of creating graphs as described above.

2.4 Hyperbolic Geometry

Hyperbolic geometry is a rich and varied field, but there are a few very specific points that need to be described as a preliminary to the rest of the paper. First off, we need to discuss the Poincare half plane model of $\mathbb{H}^2$, the equivalent of $\mathbb{R}^2$ under hyperbolic geometry. This represents hyperbolic geometry using a specific metric applied to the upper half of a plane. In the upper-half plane,

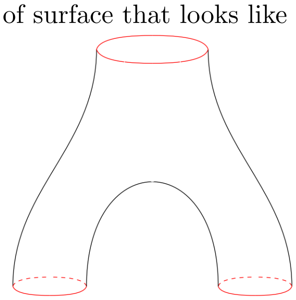
[1]This property is that they are Anosov: defining this property is beyond the scope of this paper.

the metric for any horizontal line is similar to the Euclidean metric, although with a multiple of $1/y$, where y is the y-coordinate of the horizontal line, while the metric for vertical is, for a point with coordinates (c, a) and (c, b), $\ln(b/a)$. A general notion of distance in the hyperbolic plane is difficult to express algebraically in terms of the coordinates (though it can be expressed in terms of infinitesimal distances as $ds^2 = \frac{dx^2+dy^2}{y^2}$). The x-axis is not a member of this space, but anything infinitesimally close to the x-axis is: the x-axis is essentially "at infinity." In the upper-half plane model of hyperbolic space, geodesics are therefore vertical lines and semicircles, both of which intersect with and terminate at the x-axis. The isometries of the hyperbolic plane are characterized by Mobius transformations, which send a point z to $\frac{az+b}{cz+d}$. The space of all Mobius transformations is isometric to $PSL(2, \mathbb{R})$, so each matrix can be taken to be an isometry of the hyperbolic plane [8]. Therefore, the set of images of the representations within any Teichmüller space is isomorphic to a subgroup of the isometry group of the upper half-plane, $\mathrm{Isom}(\mathbb{H}^2)$. Therefore Teichmüller spaces themselves are isomorphic to a subgroup of $\mathrm{Hom}(\pi(S), \mathrm{Isom}(\mathbb{H})^2)/\mathrm{Isom}(\mathbb{H}^2)$.

Now, we can call surfaces with constant negative curvature at any point under some Riemannian metric (which is a way of measuring distance using a smooth inner product, like with the standard Euclidean distances) on their surfaces hyperbolic manifolds. Such manifolds include a variety of torus-like shapes with extra holes, otherwise known as the genus n surfaces. A couple examples of these are displayed in the next section.

2.5 Fenchel-Nielsen Coordinates and Representations

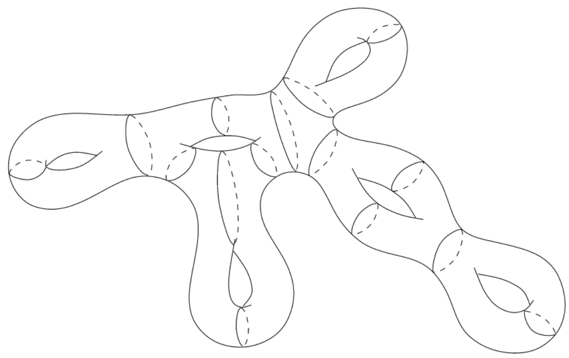
To understand how we find the desired matrix representations, we start with the theory of Fenchel-Nielsen Coordinates[2]. They are a coordinate space, created for parameterizing all the points in the Teichmüller space of the given figure. These work essentially by decomposing the surface into pairs of pants (a type of surface that looks like this:



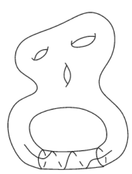
). Decomposing a generic example of a closed hyperbolic surface into a pair of

[2]All graphics in this section are from the cited paper by Paur, except for the pair of pants graphics, which is from the Wikimedia foundation and can be found at this URL: https://commons.wikimedia.org/wiki/File:Pair_of_pants.png

pants looks like this:



For instance, you could decompose a genus 2 surface into 2 pairs of pants, by first gluing the "legs" together, and then attaching them at the waists. You can do this with any hyperbolic manifold. These pairs of pants are specifically bounded by three geodesics, according to a hyperbolic metric on that space. The lengths of these geodesics determine the Fenchel-Nielsen coordinates. We then can choose how much to twist the ends of the pairs of pants by a certain amount, before gluing them together. This is a Dehn Twist, which works as follows. Take a geodesic perpendicular to the geodesic you're gluing together. For a rotation θ, the path of the geodesic is twisted by θ. However, for a rotation greater than 2π, the geodesic simply loops around the surface as many times as required. This is what a Dehn twist looks like:



This gives us 6 coordinates, the first three of which can only be positive (because they are lengths), and the remainder of which can be any real number. These coordinates actually fully parameterize the Teichmüller space of the surface: every representation in the Teichmüller space can be described by these coordinates. [7]

However, these coordinates are not unique. you may note that there is another way to glue the pairs of pants together to form a genus 2 surface. Specifically, you can glue all three boundary components to that of the other pair of pants. This produces a completely different set of Fenchel-Nielsen coordinates, so that the same coordinates map to a different element of the Teichmüller space. This is known as an F-N system.

A paper from Bernard Maskit, from 2012, outlines how, given a surface, coordinates in an F-N system can be mapped to a matrix group in the image of a Teichmüller space, or the Fuchsian group defined by the point in the F-N system. His procedure works for any surface and F-N system. He does this by taking the "Fuchsian model" of the surface, with a hyperbolic metric, meaning a quotient of the upper half plane by a Fuchsian group. In other words, he finds a Fuchsian group such that this quotient has curve lengths and twists given by the Fenchel-Nielsen coordinates.

A more detailed account of Maskit's algorithm can be taken from how he represents pairs of pants and how he glues them. One can draw a representation of this Fuchsian group in the upper-half plane with geodesics (semicircles) which intersect with the "x-axis" at points given by the division of the x and y value in their eigenvectors. These are the geodesics that the Fuschian generators fix. Assuming all the Fenchel-Nielsen coordinates are positive (neither zero or complex), we find that for a single pair of pants, the largest one maps to an isometry a_1 that fixes the y-axis, while one smaller than or equal to that one maps to an isometry a_2 fixing a semicircle such that it is perpendicular to the unit circle where it intersects it. The other end maps to an isometry a_3 that is equal to $a_2^{-1}a_1^{-1}$ anyway. Due to the normalization rule Maskit is using, this winds up equal to $-a_2^{-1}a_1^{-1}$ the matrix representation. For the hyperbolic representation, it winds up fixing a geodesic perpendicular to a circle with radius equal to the first Fenchel-Nielsen coordinate.

If two of the legs are attached to each other, then we need a "handle-closer" d, an extra isometry that represents that fact. d as a hyperbolic isometry, maps the geodesic a_2 fixes onto the geodesic a_3 does. Therefore, the matrix corresponding to that hyperbolic isometry maps the matrix for a_2 to that for a_3 by conjugation, making it theoretically computable.

Maskit then defines conjugator matrices that map pairs of pants onto each other, by mapping their half-planes on each other so that the part where they're attached lines up. These then conjugate the final matrices for those attached boundary components. This conjugator depends on which parts of the pair of pants are attached, and in our case, it works out to be trivial. Since it's irrelevant to my results, I will not be fully explaining the conjugator, but you can read Maskit to find out more about it. [6]

2.6 A Positive Representation

Recall how the Hitchin component works: there is a representative representation of the surface group in a group G, created by fixing any discrete and faithful representation of the surface group in $SL(2,\mathbb{R})$ and then composing it with a specific map into the group G. For $SL(n,\mathbb{R})$, this representation is determined as follows. We take the matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ as a vector of two polynomials

$F = aX + cY$ and $G = bX + dY$. We then take the matrix in $SL(n+1, \mathbb{R})$ to be determined by the powers of F and G in the polynomial $(F+G)^n$. For instance, in $SL(3, \mathbb{R})$, we find the matrix is determined by $(F+G)^2 = F^2 + FG + G^2$ (ignoring coefficients), so, the columns are then determined by the coefficients of F^2, FG, and G^2. This map is displayed explicitly in the methods section.

Recent research from Aparicio-Arroyo et al. shows that there exists another type of representation that is also discrete and faithful, as well as dominated, tentatively called a "positive" representation (the name is arbitrary, and it will probably be called something else once it's studied more). Their method also uses Higgs Bundles, but their conclusion is that, given a Hitchin representation ρ_H into $SO_0(p, p-1)$ from $\pi_1(S)$, and any representation at all α of $\pi_1(S)$ in $O(q-p+1)$, there is a positive p-dominated representation of the surface group into $SO(p, q)$, which is defined by $\rho = \rho_H \otimes \det(\alpha) \bigoplus \alpha$. What this means, essentially, is that we need a representation into $SO(3, 2)$, and then we simply create a block diagonal matrix with entries being the matrix in the image of that representation, and a matrix in the image of α. Furthermore, for simplicity's sake, we can choose α to be trivial, meaning that the second matrix of the block diagonal is static. If we choose it to have determinant 1, that simplifies things even further, as we no longer have to worry about the scalar multiple. Lastly, we can even take the previous representation for the $n = 5$ case (again, displayed in the methods section) to be a representation into $SO(3, 2)$ with a change of basis, as it is very close to symmetric. However, because this would not affect the eigenvalue gaps, it's actually not necessary to change the basis at all. The connected components containing it then contain positive representations. This is big news, because in general, $SO(p, q)$ does not meet the qualifications to admit a Hitchin component or induce a maximal representation. [1]

2.7 Bending

Although Fenchel-Nielsen coordinates can be used to fully parameterize Teichmüller spaces, they don't do so for higher Teichmüller Spaces. This is less surprising than it sounds: every Hitchin component of a surface has an embedded copy of the Teichmüller space defined above, by the representations in Teichmüller space composed with a natural map, as does the space of positive representations in the same way. It stands to reason that Fenchel-Nielsen coordinates could parameterize this embedded copy of Teichmüller space, a subset of the higher Teichmüller space. This is true, but the subset they parameterize is boring and is useless for our computational test: every graph of eigenvalues, like the one proposed earlier, just gives you a diagonal 45 degree line. Therefore, another procedure is required to fully explore the Higher Teichmüller Space, or explore more interesting subspaces of it. This procedure is bending, or more specifically projective convex bending. The procedure of doing projective convex bending in 3 dimensions, along a curve, is defined below. Bending is always assumed to refer to this particular type of bending from here on forth; other types of bending, such as the hyperbolic bending that was studied first, will not be defined in this paper.

Take a surface S and a curve on the surface C. Let their fundamental groups be Π and Σ, with $\Sigma \subset \Pi$. Furthermore, we can define a representation ρ from Σ into a Lie group as above. Now, if the curve C separates the surface into two components, then you can bend in this way. We can take the two components of the surface to be S_1 and S_2 and take subsets of the fundamental group corresponding to them, Π_1 and Π_2. Then, we define a new representation ρ_B, which is equal to our original ρ on Π_1, but equal to $c\rho$ on Π_2, where c conjugates the matrix ρ gives by some matrix b. Now, in order for these two maps to agree on $\Pi_1 \cup \Pi_2$, or Σ, c must be in the centralizer of Σ. Now, ρ_B gives us a bent version of the representation. Since our matrix c is in a Lie group, we can smoothly vary the amount of bending by smoothly varying the value of c. We can increase the amount of bending by choosing c such that ρ_B is less similar to ρ. This is the circumstance in which we bend in this paper. [2] We can see this as continuously deforming the representations within the connected component of the representation variety within a given Higher Teichmüller space. Per correspondence with my thesis advisor, it is generally accepted that using projective convex bending, bending along certain curves, or even bending multiple times, can allow one to fully view every single representation in a higher Teichmüller space, although this has yet to be formally proven.

3 Methods

For simplification, and ease of applying bending, I modified the matrices that appear in Maskit. Maskit's procedure for $SL(2,\mathbb{R})$ uses two pairs of pants attached to each other at all three geodesics. But, none of these geodesics are separating, as is required to bend using the procedure given in the background. Instead, I used the model of the genus 2 surface that has the two pair of pants attached at the waist, with the legs attached to each other. I found three generators: one representing the waist, one representing one leg of the left pant, and one representing one leg of the right pant, that generate the Fuschian group. Since the waist is a separating curve, we have a generator representing a seperating curve, as desired. I also found generators representing the other legs, but these were just products of the previous generators, per Maskit. Unfortunately, I did not find a closed-form formula for the handle-closers, nor did I find a way of calculating it, so those were left out. The first two generators, which I call a_1 and a_2, were derived from Maskit's formula for a pair of pants. If $(s_1, s_2, s_3, t_1, t_2, t_3)$ are the Fenchel-Nielsen coordinates of a given point, theoretically we define σ_i and τ_i by $s_i = \sinh \sigma_i$ and and $t_i = \sinh \tau_i$. (In practice since I was moreso interested in studying bending, I just filled in 1s, as these give our matrices eigenvalues that are easily expressible in terms of e. The program is designed to use σ_i and τ_i directly, instead of the input being the Fenchel-Nielsen coordinates). Anyways, in this case,

$$a_1 = \begin{bmatrix} e^{\sigma_1} & 0 \\ 0 & e^{-\sigma_1} \end{bmatrix}, a_2 = \frac{1}{\sinh\mu}\begin{bmatrix} \sinh\mu - \sigma_2 & \sinh\sigma_2 \\ -\sinh\sigma_2 & \sinh\mu + \sigma_2 \end{bmatrix},$$

where

$$\coth\mu = \frac{\cosh\sigma_1 \cosh\sigma_2 + \cosh\sigma_3}{\sinh\sigma_1 \sinh\sigma_2}.$$

Now, the requirement for the third generator, which I'd call a_2' is that it holds fixed a coordinate geodesic with points on the "x-axis" the negative of those a_2 holds fixed, since it has the same properties as a_2 but is on the opposite "side" of the pair of pants. As it turns out,

$$a_2' = \frac{1}{\sinh\mu}\begin{bmatrix} \sinh\mu + \sigma_2 & \sinh\sigma_2 \\ -\sinh\sigma_2 & \sinh\mu - \sigma_2 \end{bmatrix}.$$

The elements in the upper left and lower right corners are swapped. I found this through trial and error, though it's simple to check that it works.

The program used is fairly simple. First it takes the three matrices described above, and then applies homomorphisms from them into the target group, whether it be $SL(3, \mathbb{R})$ or $SO(p, q)$. After this it multiplies the matrices to each other around ten thousand times to create 10000 elements of the Fuschian group. It does this by going through the array of matrices and multiplying each element successively by a_1, a_2, a_2', and their inverses, and adding these multiples to the array. It then redoes the multiplication for only the added elements. These are non-unique, not only because of the relation in the surface group, but also because the program may multiply a matrix by a_1 and then multiply that matrix by a_1^{-1}, giving us the original matrix over again. Doing so is faster then checking a matrix against every other matrix for uniqueness, though there may be other algorithms that allow checking for uniqueness to give a speed increase. It then calculates their eigenvalues (and eigenvectors, which is unnecessary), and uses this to calculate their eigenvalue gaps. It then graphs the eigenvalue gaps against one another, either in a 2-D graph for spaces with three eigenvalues, or a 4D graph for spaces with 5.

The program uses the mpmath package, instead of the standard numpy. This package is optimized for precision over speed, and is necessary due to a phenomenon that happens when the floating point numbers being used are too low precision. This leads to a "drift" leftwards and downwards. All graphs in the results section are generated using 151-bit precision.

Now, what homomorphism is used? Although it's explained in the background, the explanation would be more clear if it was written out explicitly here. The homomorphism into $SL(3,\mathbb{R})$ is given by

$$\rho : \begin{pmatrix} a & b \\ c & d \end{pmatrix} \to \begin{pmatrix} a^2 & ab & b^2 \\ 2ac & ad+bc & 2bd \\ c^2 & cd & d^2 \end{pmatrix}.$$

The homomorphism into $SL(5,\mathbb{R})$ is given by

$$\rho : \begin{pmatrix} a & b \\ c & d \end{pmatrix} \to \begin{pmatrix} a^4 & a^3b & a^2b^2 & ab^3 & b^4 \\ 4a^3c & 3a^2bc+a^3d & 2ab^2c+2a^2bd & b^3c+3ab^2d & 4b^3d \\ 6a^2c^2 & 3abc^2+3a^2cd & b^2c^2+4abcd+a^2d^2 & 3b^2cd+3abd^2 & 6b^2d^2 \\ 4ac^3 & bc^3+3ac^2d & 2bc^2d+2acd^2 & 3bcd^2+ad^3 & 4bd^3 \\ c^4 & c^3d & c^2d^2 & cd^3 & d^4 \end{pmatrix}$$

. The homomorphism into $SO(3,4)$ is given by that previous one on a block diagonal with the 2×2 identity matrix.

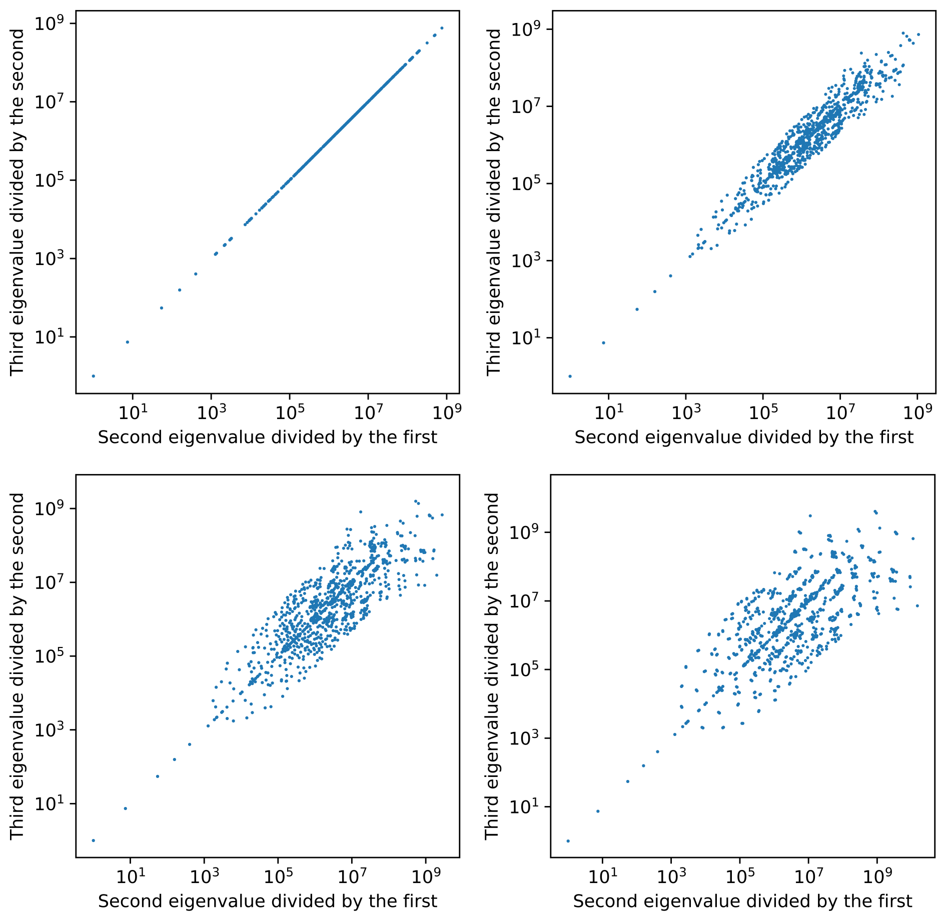
How is bending implemented? It's pretty simple. Because the image of the curve we're bending along, which is a_1, is a diagonal matrix, the matrix we conjugate with to bend is one that fixes, or commutes with, a diagonal matrix. This is any other diagonal matrix. However, all of the homomorphisms above, if you check, map a_1, our surface group generator to a matrix with at least 1 1 on the diagonal. So, we can actually bend with a larger variety of matrices, although I did not do this. The matrix that in particular is conjugated is a_2'.

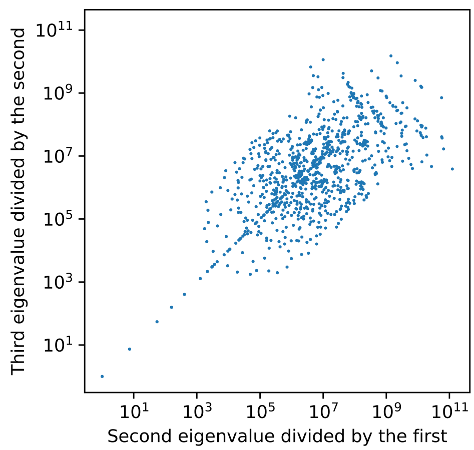
There was however, a problem, the homomorphism, as implemented, was not capable of being bent outside a subspace of the Teichmüller space that was fairly boring while continuing to give the correct behavior. This is because the matrix homomorphism to $SO(3,4)$ maps any matrix to a block diagonal matrix, consisting of 1 5 x 5 matrix and 1 2x2 matrix whose entries do not depend on the original matrix. The eigenvalues of every matrix in $SO(3,4)$ look like $\lambda_1, \lambda_2, \lambda_3, 1, \lambda_3^{-1}, \lambda_2^{-1}, \lambda_1^{-1}$. (This is because for every matrix for SO(3, 4), by definition, $A^TA = J$, where J is the diagonal matrix with entries consisting of 4 1s and 3 -1s. If A has eigenvalue λ and eigenvector v, $Av = \lambda v$. But, since $A^T = JA^{-1}$ $A^Tv = JA^{-1}v$, or $J\lambda^{-1}v$, which is $\lambda^{-1}Jv$. This means λ^{-1} is an eigenvalue of A^T, which means its an eigenvalue of A.) However, because we're creating two eigenvalues independent of our original matrix, λ_3 actually has nothing to do with the original matrix, meaning it has nothing to do with the related structures we're studying. Therefore, we get no additional information from graphing the eigenvalue gap between λ_3 and 1, and in many cases (like the case the program ran in), it may violate the bound on the eigenvalue gaps that holds for p-dominated representations. Bending cannot resolve this: the bound on the eigenvalue gap is always violated by the subgroup generated by the matrices a_1 and a_2, which remain unbent. So, the choice of homomorphism fundamentally does not work with this method. Unfortunately I didn't have time to make a different choice or adjust the method, so I went for something else.

Bending in this scenario is on some level an unknown quantity. So, I decided to use the program to investigate how bending works in spaces like $SL(3,\mathbb{R})$ and $SL(5,\mathbb{R})$, and compare them to a change in Fenchel Nielsen coordinates.

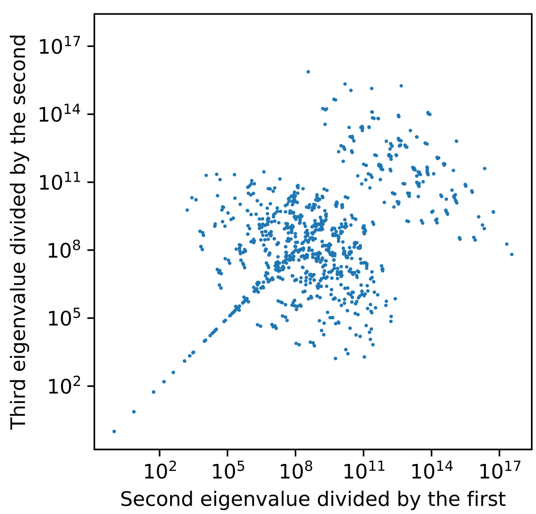
4 Results

These graphs display the eigenvalue graphs of 10005 non-unique matrices (probably a few hundred to a few thousand unique matrices) in Maskit's representation of the matrices in the image of Teichmüller space given in the methods (meaning, excluding the handle-closers), with Fenchel-Nielsen coordinates all 1s except when otherwise stated. The generator matrices were sent to $SL(3,\mathbb{R})$, and the matrix a_2' was bent. The bending here was done with the matrix with entries on the diagonal $1, 2, 1/2$ raised to the 0th, .5th, 1st, 1.5th and 2nd powers.

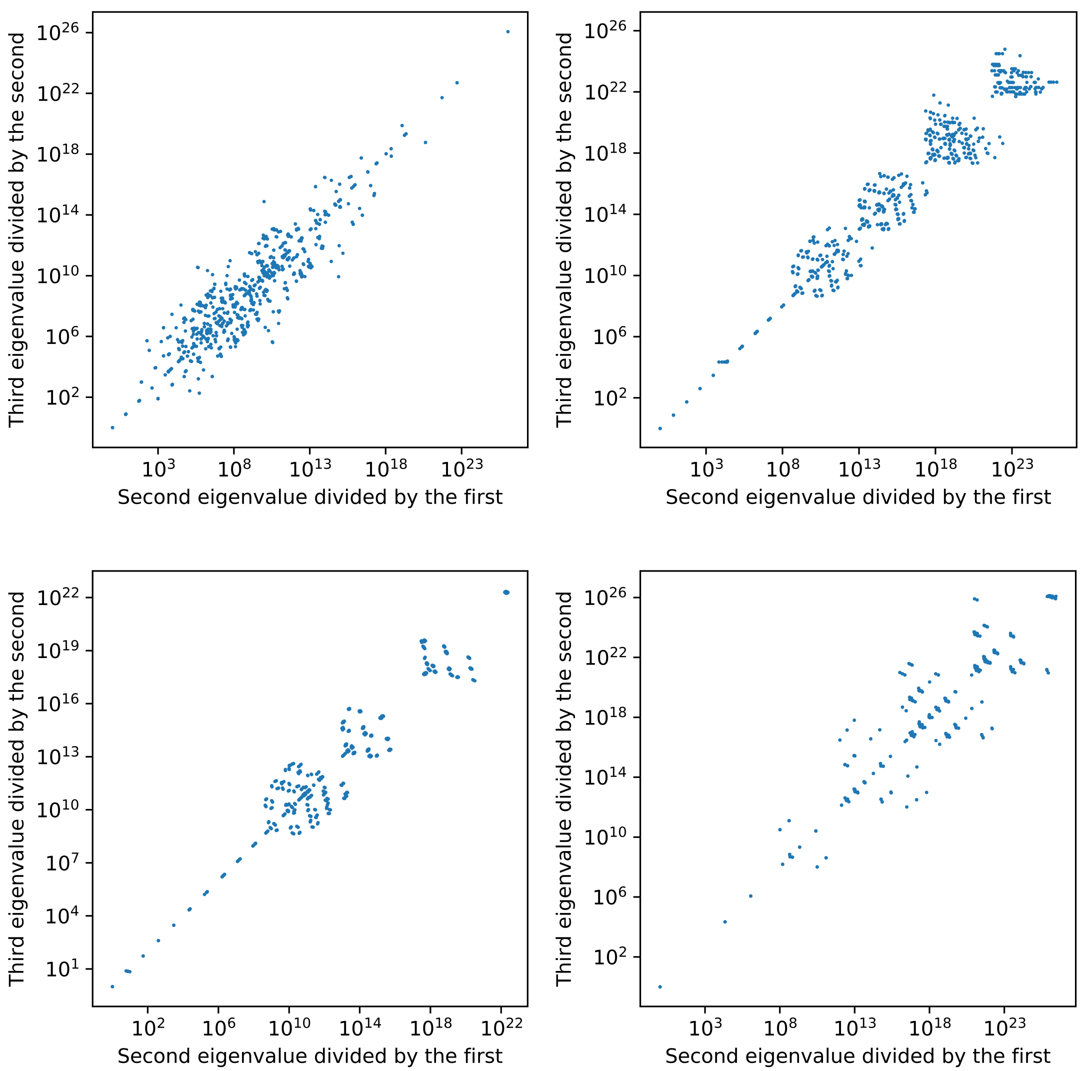




The effects of bending on this group are clear. One can see that the graphs look like a spoon: starting with a straight line, then having bending applied. The further the matrix group is bent, the more the bending curve spreads. This matrix correctly bends the group further and further. In the last two diagrams, you can see that a different characteristic behavior is displayed by points around $(20, 20)$, or so. Indeed if you bend further, which is harder on the computer and leads to more artifacting, you find that the two sections has separated completely with something that looks like the gap between two parts of a hyperbola in between. Here is an extreme bending of roughly our matrix to the 6th power, demonstrating what happens with this bending.

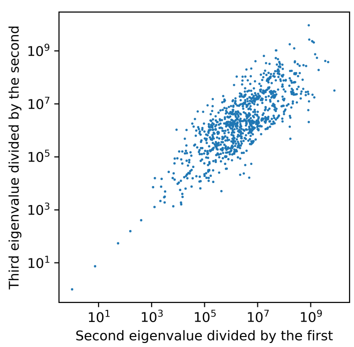


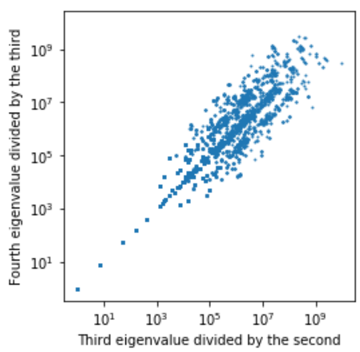
The limit of bending, then, might be two completely seperated graphs. It is impossible to truly see due to the small number of points and the tendency of the computer to produce artifacts in large bendings, to the point where bendings using our matrix to the 20th power or more throw unresolvable errors. However, bendings using our matrix to the 20th power look similar enough to bendings using our matrix to the 6th, supporting our theory that this is what the limit of bending looks like. While the components may get wider, they're still along the same line. We now compare bending to a change in Fenchel Nielsen coordinates, with a constant bending of our matrix to the 2nd power. The four graphs below show us the effect of raising the first Fenchel Nielsen coordinate from 1 to 5, the second from 1 to 5, the third from 1 to 5, and all three from one to five. This is what is produced:

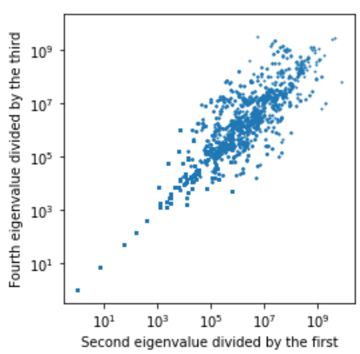


Although raising each Fenchel Nielsen coordinate changes the group in a diverse way, particularly the first as it affects a_1, they all seem to trend towards a series of explosions, similar to the two in the extreme bending graph. Given the larger bounds on this graph, we can speculate that these may represent a "zoomed out" view of the previous graphs. In this case, we can speculate that the limit of bending, were we capable of analyzing infinite matrices, is infinity of these "explosions."

Why this is the limit of bending might have something to do with the structure of the bent group. In this group, only a_2' is bent. So, it might be that the group elements are bent by a_2', then their squares are, then their cubes are, and so on and so forth. This is an unsatisfactory explanation, however, and I fail to find a satisfactory one. Additionally, you can see that bending also works in $SL(5, \mathbb{R})$ using the matrix with diagonal entries $(1, 2, 1/2, 3/2, 2/3)$. Displayed is the bending by the square of that matrix for the first three eigenvalue gaps.







These maps consist of the 1st eigenvalue gap vs the 2nd ($\frac{\lambda_2}{\lambda_1}$ vs $\frac{\lambda_3}{\lambda_2}$), the second vs the third, and the first vs the third. These graphs support that you can vary the amount of bending by varying the entries on the diagonal, as the third eigenvalue gap appears to have a slightly different behavior, especially when plotted against the first. A specific correspondence between how far each entry is from 1 and the amount of bending in different dimensions may exist.

5 Discussion

Unfortunately, my original plan (to create a program capable of investigating every element of a higher Teichmüller space) did not work out. However, you may notice the vast majority of the thesis is still focused on building up to this: I feel like this would have been possible if not for time constraints. Some things that might be helpful are porting my code, which is in python, an inefficient langauge, to C++, a much more efficient language. This would require a quad-precision and a matrix library to replace mpmath. I had started doing a C++ port using Eigen (a matrix library for C++) and a quad-precision library I found online, but this port was abandoned due to time constraints. It is my belief that were this completed, the port would be capable of investigating 10 million group elements or more, as opposed to the 10,000 the program currently implements. There may also be some mathematical way to circumvent the large precision requirements of this program so it can run faster and more extreme bendings may be displayed. Additionally, if a way of easily calculating the handle closers could be found, or if a way of calculating them could be computationally implemented, then a larger subspace of the Teichmüller space would able to be found. Lastly, there is a real chance that my code is erroneous, as even if it did have an error, there's still a chance it'd produce bounded graphs in $SL(3, \mathbb{R})$ as it did, although there is a false positive rate like this in every experiment. However, the largest problem with my program was a fundamental mathematical one. Specifically, the "positive representation" used was not capable of being bent outside a small subset of the Teichmüller space. Therefore, it may be possible to use a different positive representation, and bend using that, so that the procedure laid out in this paper works. Additionally, it may be possible to bend along a different curve in order to explore every element in the space. You would then just need to test a variety of bendings and Fenchel-Nielsen coordinates, and see if any of the graphs show that the relevant representation violates the p-dominated property.

As for the results I did get, I believe they indicate how bending can be used in this test, and what effects it has. As I said earlier, it is a generally accepted that this convex projective bending can be used to explore every element in a Higher Teichmüller space. This paper shows, at least, one property of the representations a certain bending produces. Convex projective bending is a relatively new concept, to the extent that the Ballas and Marquis paper ([2]) which establishes its properties, was written in 2016 and revised this year, in 2020. This very clearly and graphically displays how it, when done right, mantains the p-dominated property, and therefore the Anosov property, of the representation. This shows how we can explore the matrices in the image of representations in a higher Teichmuller space, ultimately allowing us to explore many discrete subgroups of $SL(n, \mathbb{R})$. This is meant to clarify how bending works within higher Teichmüller spaces. Additionally, it displays what happens at the limits of bending. A proper explanation for this eludes me, but it's interesting phenomena regardless. Still, hopefully this exploration can clarify some things about the proposed work described above.

6 APPENDIX A: My Code

```
import matplotlib.pyplot as plt
from mpmath import mp
from mpl_toolkits.mplot3d import Axes3D
import math
mp.prec = 151
mp.pretty = False
def tosl3r(ps2):
    m = ps2.tolist()
    a = m[0][0]
    b = m[0][1]
    c = m[1][0]
    d = m[1][1]
    #Uncomment for SL(3, R):
    #return mp.matrix([[a**2, a ^b, b**2], [2*a*c, a*d+b*c, 2*b*d], [c**2, c
        ↪ *d, d**2]])
    #Uncomment for SO(3, 4) (doesn't work properly)
    #return mp.matrix([[(a**4), (a**3)*b, (a**2)*(b**2), (a)*(b**3), b**4,
        ↪ 0, 0], [4*(a**3)*c, 3*(a**2)*b*c + (a**3)*d, 2*a*(b**2)*c + 2*(
        ↪ a**2)*b*d, (b**3)*c + 3*a*(b**2)*d, 4*(b**3)*d, 0, 0], [6*(a
        ↪ **2)*(c**2), 3*a*b*(c**2) + 3*(a**2)*c*d, (b**2)*(c**2) + 4*a
        ↪ *b*c*d + (a**2)*(d**2), 3*(b**2)*c*d + 3*a*b*(d**2), 6*(b**2)
        ↪ *(d**2), 0, 0], [4*a*(c**3), (b)*(c**3) + 3*a*(c**2)*d, 2*b*(c
        ↪ **2)*d + 2*a*c*(d**2), 3*b*c*(d**2)+ a*(d**3), 4*b*(d**3), 0,
        ↪  0], [(c**4), (c**3)*d, (c**2)*(d**2), (c)*(d**3), d**4, 0, 0], [0,
        ↪ 0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 0, 1]])
    #Uncomment for SL(5, R):
    return mp.matrix([[(a**4), (a**3)*b, (a**2)*(b**2), (a)*(b**3), b**4],
        ↪ [4*(a**3)*c, 3*(a**2)*b*c + (a**3)*d, 2*a*(b**2)*c + 2*(a**2)*
        ↪ b*d, (b**3)*c + 3*a*(b**2)*d, 4*(b**3)*d], [6*(a**2)*(c**2), 3*a
        ↪ *b*(c**2) + 3*(a**2)*c*d, (b**2)*(c**2) + 4*a*b*c*d + (a**2)
        ↪ *(d**2), 3*(b**2)*c*d + 3*a*b*(d**2), 6*(b**2)*(d**2)], [4*a*(c
        ↪ **3), (b)*(c**3) + 3*a*(c**2)*d, 2*b*(c**2)*d + 2*a*c*(d**2),
        ↪ 3*b*c*(d**2)+ a*(d**3), 4*b*(d**3)], [(c**4), (c**3)*d, (c**2)*(
        ↪ d**2), (c)*(d**3), d**4]])
ga = [1, 1,1 ,1 ,1, 1] #ga contains our sigmas and taus
ben = mp.diag([1, 2**.1, ((1/2)**(.1)), ((3/2)**(.1)), ((2/3)**(.1))])
u = mp.atanh(1/((mp.cosh(ga[0])*mp.cosh(ga[1]) + mp.cosh(ga[2]))/(mp.sinh
    ↪ (ga[0])*mp.sinh(ga[1]))))
```

```
v = mp.atanh(1/((mp.cosh(ga[0])*mp.cosh(ga[2]) + mp.cosh(ga[3]))/(mp.sinh
    ↪ (ga[0])*mp.sinh(ga[2]))))
a = tosl3r(mp.matrix([[mp.exp(ga[0]), 0], [0, mp.exp(-ga[0])]]))
b = tosl3r(mp.matrix([[(1/mp.sinh(u))*mp.sinh(u - ga[1]), (1/mp.sinh(u))*
    ↪ mp.sinh(ga[1])], [(1/mp.sinh(u))*(-mp.sinh(ga[1])), (1/mp.sinh(u))*
    ↪ mp.sinh(u + ga[1])]]))
for n in range(1, 21):
    c = (ben**n)*tosl3r(mp.matrix([[(1/mp.sinh(u))*mp.sinh(u + ga[1]), (1/
        ↪ mp.sinh(u))*mp.sinh(ga[1])], [(1/mp.sinh(u))*(-mp.sinh(ga[1])),
        ↪ (1/mp.sinh(u))*mp.sinh(u - ga[1])]]))*(ben**(-n))
    matrixarray = [a, b, c]
    eff = 0
    points = 1000
    while len(matrixarray)<points:
        to_eff = len(matrixarray)
        for i in range(eff, len(matrixarray)):
            matrixarray.append(matrixarray[i]* a)
            matrixarray.append(matrixarray[i]* b )
            matrixarray.append(matrixarray[i]* c)
            matrixarray.append(matrixarray[i]* a**-1)
            matrixarray.append(matrixarray[i]* b**-1)
            matrixarray.append(matrixarray[i]* c**-1)
            if len(matrixarray)>points:
                break
        eff = to_eff
    print("check 1, array len:"+ str(len(matrixarray))) #checks are to see
        ↪ where the program is having trouble if it fails to terminate
    eiggaps12array = []
    eiggaps23array = []
    eiggaps34array = []
    for i in range(0, len(matrixarray)):
        try:
            evals, el = mp.eig(matrixarray[i])
        except RuntimeError:
            print ("oops")
            continue
        evals = sorted(evals, key = abs)
        eiggaps12array.append(math.log(float(abs(evals[1]))/float(abs(evals
            ↪ [0]))))
        eiggaps23array.append(math.log(float(abs(evals[2]))/float(abs(evals
            ↪ [1]))))
        eiggaps34array.append(math.log(float(abs(evals[3]))/float(abs(evals
            ↪ [2]))))
    print("check 2")
    plt.scatter(eiggaps12array, eiggaps23array, s = 1)
    plt.show()
```

```
#code below, and the whole eiggaps34 array, can be deleted if youre not
    ↪ doing SL(5, R) or SO(3, 4)
plt.scatter(eiggaps23array, eiggaps34array, s = 1)
plt.show()
plt.scatter(eiggaps12array, eiggaps34array, s = 1)
plt.show()
```

7 APPENDIX B: Stray Remark

First off, due to the fact that the name Teichmüller is used in this paper around 50 times, I feel obligated to state somewhere that Oscar Teichmüller, the creator of Teichmüller theory, was, in fact, a proud member of the Nazi party of Germany. Personally, I often tend to unconsciously end up revering the mathematicians whose work I study, so this is intended as a corrective to that.

References

[1] Marta Aparicio-Arroyo, Steven Bradlow, Brian Collier, Oscar García-Prada, Peter B. Gothen, and André Oliveira. SO(p, q)-higgs bundles and higher teichmüller components. *Inventiones mathematicae*, 218(1):197–299, Jun 2019.

[2] Samuel A. Ballas and Ludovic Marquis. Properly convex bending of hyperbolic manifolds, 2020.

[3] Jairo Bochi, Rafael Potrie, and Andrés Sambarino. Anosov representations and dominated splittings. *Journal of the European Mathematical Society*, 21(11):3343–3414, Jul 2019.

[4] Olivier Guichard and Anna Wienhard. Positivity and higher teichmüller theory, 2018.

[5] R Andrew Kreek. Surface groups. 2007.

[6] Bernard Maskit. Matrices for fenchel-nielsen coordinates. *RECON no. 20010088230.; Annales Academiae Scientiarum Fennicae: Mathmatica*, 26(2):267–304, 2001.

[7] Kathy Paur. The fenchel-nielsen coordinates of techmuller space. *MIT UJM*, 1:149–154, 1999.

[8] Kathy Snyder. Isometries of the hyperbolic plane. *Research Experience for Undergraduates, University of Chicago*, 1:13, 2009.

[9] Anna Wienhard. An invitation to higher teichmüller theory, 2018.